PR #51668 完整报告

vllm-project/vllm

Bump Transformers version to 5.15.0

合并时间：2026-08-12 00:00

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51668

执行摘要

- 一句话：升级 Transformers 至 5.15.0 并适配其 API 变化
- 推荐动作：值得快速浏览：重点看 multimodal.py 的兼容层写法（getattr 回退 + torch.isin）与 _HeterogeneousConfig 的初始化顺序调整，这两个模式可复用于其他向 transformers 5.15 迁移的适配。若团队维护 Transformers 建模后端或多模态 pipeline，建议精读源码片段并关注 5.15 的 release notes。

功能与动机

PR 正文与关联 Issue 均为空，动机主要从变更内容推断：跟随 HuggingFace transformers 5.15.0 发布，将 CI 测试依赖从 5.14.1 升级到 5.15.0，以保持测试环境与上游对齐。5.15 引入了两个影响 vLLM 的 API 变化——processor 的 image_token_ids 集合属性（替代单一 image_token_id）与异构配置初始化校验——因此同步做了小范围适配；此外 test_mla 改用 tiny 模型，减少测试资源消耗。

实现拆解

1. 依赖版本升级：将 requirements/test/cuda.in、rocm.in、xpu.in 中的 transformers==5.14.1 升为 transformers==5.15.0，并同步更新编译后的锁定文件 cuda.txt、rocm.txt、xpu.txt、cpu.txt、nightly-torch.txt，确保 CUDA、ROCm、XPU、CPU 各 CI 环境版本一致。
2. 多模态 placeholder 兼容层：在 vllm/model_executor/models/transformers/multimodal.py 的 _apply_vision 中，is_embed 判断从单一 image_token_id 改为优先读取 hf_processor.image_token_ids 集合（Transformers >= 5.15 提供），旧版本回退到 image_token_id，并使用 torch.isin 批量判断，适配 5.15 支持多 image token id 的模型。
3. 异构配置测试桩修复：tests/config/test_model_arch_config.py 的 _HeterogeneousConfig 需要把 _per_layer 的赋值挪到 super().__init__() 之前，因为新版 Transformers 在初始化时会校验 per_layer_config，顺序不对会直接抛错。
4. MLA 测试模型替换：tests/models/transformers/test_backend.py 的 test_mla 从 get_model("DeepseekV2ForCausalLM") 改为 hmellor/tiny-random-DeepseekV2ForCausalLM，用 tiny 模型降低测试资源占用和远程下载失败风险。
5. 全量 CI 验证：作者多次触发 /ci run all，第一轮 CI 因不稳定步骤失败后重新触发，最后于新 commit 上通过。

关键文件：

- vllm/model_executor/models/transformers/multimodal.py（模块 多模态处理；类别 source；类型 data-contract；符号 _apply_vision）：唯一源码改动，适配 Transformers 5.15 多 image token id 契约，是多模态 placeholder 判定的核心兼容层。
- tests/config/test_model_arch_config.py（模块 模型配置；类别 test；类型 test-coverage；符号 _HeterogeneousConfig, test_transformers_heterogeneous_config_is_resolved_per_layer）：测试桩 _HeterogeneousConfig 的初始化顺序调整直接反映 Transformers 5.15 对异构配置的校验行为变化，是适配的核心测试变更。
- tests/models/transformers/test_backend.py（模块 后端测试；类别 test；类型 test-coverage；符号 test_mla）：MLA 测试改用个人账号下的 tiny 模型，降低测试资源与远程依赖风险，是测试配套的重要调整。
- requirements/test/cuda.in（模块 依赖管理；类别 test；类型 configuration）：CUDA 测试依赖入口文件，transformers 版本从 5.14.1 升到 5.15.0。
- requirements/test/rocm.in（模块 依赖管理；类别 test；类型 configuration）：ROCm 测试依赖入口文件，transformers 版本同步升级。
- requirements/test/xpu.in（模块 依赖管理；类别 test；类型 configuration）：XPU 测试依赖入口文件，transformers 版本同步升级。
- requirements/test/cuda.txt（模块 依赖管理；类别 docs；类型 documentation）：CUDA 锁定文件，反映升级后的完整依赖解析结果。

关键符号：_apply_vision, _HeterogeneousConfig.init, test_mla

关键源码片段

vllm/model_executor/models/transformers/multimodal.py

唯一源码改动，适配 Transformers 5.15 多 image token id 契约，是多模态 placeholder 判定的核心兼容层。

```
def _apply_vision(
    self,
    prompt_ids: list[int],
    processed_data: "BatchFeature",
    mm_items: MultiModalDataItems,
    hf_processor_mm_kwargs: Mapping[str, object],
    mm_token_type_ids: torch.Tensor | None,
) -> dict[str, list[PlaceholderRange]]:
    # Placeholder 定位依赖 token type id，缺失时放弃而不是猜测
    if mm_token_type_ids is None:
        return {}

    hf_processor = self.info.get_hf_processor(**hf_processor_mm_kwargs)

    # 从 token type ids 推断 vLLM 风格的 placeholder，按输入 `mm_data` 拆分
    mm_positions = torch.where(mm_token_type_ids == 1)[1]
    images = mm_items.get_items("image", ImageProcessorItems)
```

```
    image_sizes = []
    for item_idx in range(len(images)):
        image_size = images.get_image_size(item_idx)
        image_sizes.append((image_size.height, image_size.width))

    mm_tokens_per_modality = hf_processor._get_num_multimodal_tokens(
        image_sizes=image_sizes,
        **self.info.ctx.get_merged_mm_kwargs({}),
    )

    mm_placeholders: dict[str, list[PlaceholderRange]] = {}
    split_sizes = mm_tokens_per_modality["num_image_tokens"]
    if split_sizes:
        # Transformers 5.15.0 起 processor 提供 `image_token_ids` 集合，
        # 旧版本只有单一 `image_token_id`，这里做兼容回退
        image_token_ids = getattr(hf_processor, "image_token_ids", None)
        if image_token_ids is None:
            # Transformers < 5.15.0
            image_token_ids = [hf_processor.image_token_id]
        image_token_ids = torch.tensor(
            [i for i in image_token_ids if i is not None]
        )

        chunked_mm_positions = torch.split(mm_positions, split_sizes)
        mm_tokens = torch.tensor(prompt_ids)[mm_token_type_ids[0].bool()]
        chunked_mm_tokens = torch.split(mm_tokens, split_sizes)
        ranges = [
            PlaceholderRange(
                offset=positions[0].item(),
                length=positions.shape[0],
                # 5.15 允许模型有多个 image token id，改用 `torch.isin` 判断
                is_embed=torch.isin(mm_tokens, image_token_ids),
            )
            for positions, mm_tokens in zip(
                chunked_mm_positions, chunked_mm_tokens
            )
        ]
        mm_placeholders = {"image": ranges}

    processed_data["num_image_patches"] = torch.tensor(
        mm_tokens_per_modality["num_image_patches"]
    )
    return mm_placeholders
```

tests/config/test_model_arch_config.py

测试桩 _HeterogeneousConfig 的初始化顺序调整直接反映 Transformers 5.15 对异构配置的校验行为变化，是适配的核心测试变更。

```
class _HeterogeneousConfig(PretrainedConfig):
```

```
    """无独立 convertor 的异构配置，用于模拟 Transformers 的 per-layer 接口。

    镜像 vLLM 用到的部分：per-layer 配置是浅拷贝后应用变化属性，
    并剥离异构标记，避免递归展开。
    """

    is_heterogeneous = True

    def __init__(self, per_layer: dict[str, list], **kwargs):
        # 必须在 `super().__init__` 之前设置：新版 Transformers 会在
        # 初始化时校验 `per_layer_config`，而该属性依赖 `_per_layer`
        self._per_layer = per_layer
        super().__init__(**kwargs)

    @property
    def per_layer_config(self) -> list[PretrainedConfig]:
        layers = []
        for i in range(self.num_hidden_layers):
            layer = copy(self)
            layer.is_heterogeneous = False
            for name, values in self._per_layer.items():
                setattr(layer, name, values[i])
            layers.append(layer)
        return layers
```

评论区精华

本 PR 没有实质性的 reviewer 技术讨论：claude[bot] 因 PR 来自 fork 而跳过自动 review，作者 hmellor 则是通过对 issue 区发 /ci run all、/ci retry、/ci run 驱动 CI。第一轮 Buildkite CI（#83137）出现不稳定失败且因缺少稳定 step key 无法重试，第二次触发（#83304）在最终 commit 上通过。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 依赖升级影响面：requirements/test 下所有平台锁定文件同步升到 5.15.0，若该版本对某些模型（如 MiniCPMV）有已知回归，而之前的 cap 仅作用于 HF runner（见 PR#45042），可能导致其他平台 CI 暴露新问题。
  - 多模态判定语义变化：multimodal.py 中 is_embed 从 " 等于单一 image_token_id" 变为 " 属于 image_token_ids 集合 "，若某个 processor 的 image_token_ids 包含意料之外的 token，placeholder 的 embed 标记可能变化；回退分支保证 < 5.15 行为不变。
  - 测试桩脆弱性：_HeterogeneousConfig 依赖 transformers 在 __init__ 时校验 per_layer_config 的行为，属于内部实现细节，未来版本可能再次调整。
  - 外部资源依赖：test_mla 改用个人账号 hmellor/tiny-random-DeepseekV2ForCausalLM，若模型被删除或更新，测试会受影响。

- 影响：影响集中在测试与 CI 基础设施：CUDA、ROCm、XPU、CPU 的测试环境依赖版本统一到 5.15.0；vllm/model_executor/models/transformers/multimodal.py 的改动会在使用 Transformers processor 的多模态模型上生效，但已兼容旧版本，不改变 vLLM 运行时行为；对团队而言，该 PR 是跟随 transformers 5.x 的常规升级节奏，后续 5.15 引入的其他 API 变化可能还需要跟进。
- 风险标记：依赖锁升级波及全平台 CI，多模态占位符判定语义变化，测试依赖第三方远端模型，适配代码依赖 Transformers 内部行为

关联脉络

- PR #45042 [CI] Restore MiniCPMV transformers cap, scoped to HF runner only: 同为 transformers 版本管理相关：该 PR 恢复 MiniCPMV 的 transformers 上限并限定 HF runner，与本 PR 的全局版本升级在依赖面相互影响。